

IN1000

lese fra og skrive data til filer

Innlesing fra fil

Innlesing fra fil

- mye større og mer spennende data enn fra tastatur

Innlesing fra fil

- mye større og mer spennende data enn fra tastatur
- slipper å taste inn hver gang man kjører

Innlesing fra fil

- mye større og mer spennende data enn fra tastatur
- slipper å taste inn hver gang man kjører
- innlesing fra tekstfiler er veldig rett fram i Python

Innlesing fra fil

- mye større og mer spennende data enn fra tastatur
- slipper å taste inn hver gang man kjører
- innlesing fra tekstfiler er veldig rett fram i Python
- en tekstfil åpnet i Python er en samling av linjer (tekster)

Innlesing fra fil

- mye større og mer spennende data enn fra tastatur
- slipper å taste inn hver gang man kjører
- innlesing fra tekstfiler er veldig rett fram i Python
- en tekstfil åpnet i Python er en samling av linjer (tekster)
- kan dermed iterere gjennom linjer i filen med ei for-løkke

Innlesing fra fil

- Først åpne en fil (*biblioteksfunksjonen open lager et fil-objekt*):
`min_fil = open("mittFilNavn.txt")`

Innlesing fra fil

- Først åpne en fil (*biblioteksfunksjonen open lager et fil-objekt*):

```
min_fil = open("mittFilNavn.txt")
```

- Deretter iterere gjennom hver linje i filen:

```
min_fil = open("mittFilNavn.txt")
```

```
for linje in min_fil:
```

Innlesing fra fil

- Først åpne en fil (*biblioteksfunksjonen open lager et fil-objekt*):

```
min_fil = open("mittFilNavn.txt")
```

- Deretter iterere gjennom hver linje i filen:

```
min_fil = open("mittFilNavn.txt")
```

```
for linje in min_fil:
```

- Inni for-løkka, gjør noe med linja

```
min_fil = open("mittFilNavn.txt")
```

```
for linje in min_fil:
```

```
    print("Her fant jeg: " + linje)
```

a
ability
able
about
above
accept
according
account
across
act
action
activity
actually
add
address
administration
admit
adult
affect
after
again
against
age
agency
agent
ago
agree
agreement
ahead
air
all
allow
almost
alone
along
already
also
although
always
American
among
amount
analysis
and
animal
another
answer
any
anyone

U:--- 1000eng0rd.txt

Beginning of buffer

a
ability
able
about
above
accept
according
account
across
act
action
activity
actually
add
address
administration
admit
adult
affect
after
again
against
age
agency
agent
ago
agree
agreement
ahead
air
all
allow
almost
alone
along
already
also
although
always
American
among
amount
analysis
and
animal
another
answer
any
anyone

U:--- 1000eng0rd.txt

Beginning of buffer

```
nyEngelskListe = []  
innfil = open("1000eng0rd.txt")  
for linje in innfil:  
    nyEngelskListe.append(linje)  
  
print(nyEngelskListe)  
print("Antall ord i lista var", len(nyEngelskListe))  
print("Takk for nå")
```

innlesing01.py

```
\n', 'pressure\n', 'pretty\n', 'prevent\n', 'price\n', 'private\n', 'probably\n', 'problem\n', 'process\n', 'pr  
oduce\n', 'product\n', 'production\n', 'professional\n', 'professor\n', 'program\n', 'project\n', 'property\n',  
'protect\n', 'prove\n', 'provide\n', 'public\n', 'pull\n', 'purpose\n', 'push\n', 'put\n', 'quality\n', 'quest  
ion\n', 'quickly\n', 'quite\n', 'race\n', 'radio\n', 'raise\n', 'range\n', 'rate\n', 'rather\n', 'reach\n', 're  
ad\n', 'ready\n', 'real\n', 'reality\n', 'realize\n', 'really\n', 'reason\n', 'receive\n', 'recent\n', 'recentl  
y\n', 'recognize\n', 'record\n', 'red\n', 'reduce\n', 'reflect\n', 'region\n', 'relate\n', 'relationship\n', 'r  
eligious\n', 'remain\n', 'remember\n', 'remove\n', 'report\n', 'represent\n', 'Republican\n', 'require\n', 'res  
earch\n', 'resource\n', 'respond\n', 'response\n', 'responsibility\n', 'rest\n', 'result\n', 'return\n', 'revea  
l\n', 'rich\n', 'right\n', 'rise\n', 'risk\n', 'road\n', 'rock\n', 'role\n', 'room\n', 'rule\n', 'run\n', 'safe  
\n', 'same\n', 'save\n', 'say\n', 'scene\n', 'school\n', 'science\n', 'scientist\n', 'score\n', 'sea\n', 'seaso  
n\n', 'seat\n', 'second\n', 'section\n', 'security\n', 'see\n', 'seek\n', 'seem\n', 'sell\n', 'send\n', 'senior  
\n', 'sense\n', 'series\n', 'serious\n', 'serve\n', 'service\n', 'set\n', 'seven\n', 'several\n', 'sex\n', 'sex  
ual\n', 'shake\n', 'share\n', 'she\n', 'shoot\n', 'short\n', 'shot\n', 'should\n', 'shoulder\n', 'show\n', 'sid  
e\n', 'sign\n', 'significant\n', 'similar\n', 'simple\n', 'simply\n', 'since\n', 'sing\n', 'single\n', 'sister  
\n', 'sit\n', 'site\n', 'situation\n', 'six\n', 'size\n', 'skill\n', 'skin\n', 'small\n', 'smile\n', 'so\n', 'so  
cial\n', 'society\n', 'soldier\n', 'some\n', 'somebody\n', 'someone\n', 'something\n', 'sometimes\n', 'son\n',  
'song\n', 'soon\n', 'sort\n', 'sound\n', 'source\n', 'south\n', 'southern\n', 'space\n', 'speak\n', 'special\n',  
'specific\n', 'speech\n', 'spend\n', 'sport\n', 'spring\n', 'staff\n', 'stage\n', 'stand\n', 'standard\n', 's  
tar\n', 'start\n', 'state\n', 'statement\n', 'station\n', 'stay\n', 'step\n', 'still\n', 'stock\n', 'stop\n', 's  
tore\n', 'story\n', 'strategy\n', 'street\n', 'strong\n', 'structure\n', 'student\n', 'study\n', 'stuff\n', 's  
tyle\n', 'subject\n', 'success\n', 'successful\n', 'such\n', 'suddenly\n', 'suffer\n', 'suggest\n', 'summer\n',  
'support\n', 'sure\n', 'surface\n', 'system\n', 'table\n', 'take\n', 'talk\n', 'task\n', 'tax\n', 'teach\n', 't  
eacher\n', 'team\n', 'technology\n', 'television\n', 'tell\n', 'ten\n', 'tend\n', 'term\n', 'test\n', 'than\n',  
'thank\n', 'that\n', 'the\n', 'their\n', 'them\n', 'themselves\n', 'then\n', 'theory\n', 'there\n', 'these\n',  
'they\n', 'thing\n', 'think\n', 'third\n', 'this\n', 'those\n', 'though\n', 'thought\n', 'thousand\n', 'threa  
t\n', 'three\n', 'through\n', 'throughout\n', 'throw\n', 'thus\n', 'time\n', 'to\n', 'today\n', 'together\n', 't  
onight\n', 'too\n', 'top\n', 'total\n', 'tough\n', 'toward\n', 'town\n', 'trade\n', 'traditional\n', 'training  
\n', 'travel\n', 'treat\n', 'treatment\n', 'tree\n', 'trial\n', 'trip\n', 'trouble\n', 'true\n', 'truth\n', 'tr  
y\n', 'turn\n', 'TV\n', 'two\n', 'type\n', 'under\n', 'understand\n', 'unit\n', 'until\n', 'up\n', 'upon\n', 'u  
s\n', 'use\n', 'usually\n', 'value\n', 'various\n', 'very\n', 'victim\n', 'view\n', 'violence\n', 'visit\n', 'v  
oice\n', 'vote\n', 'wait\n', 'walk\n', 'wall\n', 'want\n', 'war\n', 'watch\n', 'water\n', 'way\n', 'we\n', 'wea  
pon\n', 'wear\n', 'week\n', 'weight\n', 'well\n', 'west\n', 'western\n', 'what\n', 'whatever\n', 'when\n', 'whe  
re\n', 'whether\n', 'which\n', 'while\n', 'white\n', 'who\n', 'whole\n', 'whom\n', 'whose\n', 'why\n', 'wide\n',  
'wife\n', 'will\n', 'win\n', 'wind\n', 'window\n', 'wish\n', 'with\n', 'within\n', 'without\n', 'woman\n', 'w  
onder\n', 'word\n', 'work\n', 'worker\n', 'world\n', 'worry\n', 'would\n', 'write\n', 'writer\n', 'wrong\n', 'y  
ard\n', 'yeah\n', 'year\n', 'yes\n', 'yet\n', 'you\n', 'young\n', 'your\n', 'yourself']
```

Antall ord i lista var 1000

Takk for nå

>

`\n` er *ett* tegn som skrives ut som linjeskift

```
print("Her kan vi se hvordan '\\n' virker ved utskrift:")  
print("Her\n kan \nvi\nse \nhvordan '\n' virker ved ut\nskrift")
```

`\n` er *ett* tegn som skrives ut som linjeskift

```
print("Her kan vi se hvordan '\\n' virker ved utskrift:")  
print("Her\n kan \nvi\nse \nhvordan '\\n' virker ved ut\nskrift")
```

```
[2020_uke06 > python3 linjeskift.py  
Her kan vi se hvordan '\\n' virker ved utskrift:  
Her  
 kan  
vi  
se  
hvordan '  
' virker ved ut  
skrift  
[2020_uke06 >
```

a
ability
able
about
above
accept
according
account
across
act
action
activity
actually
add
address
administration
admit
adult
affect
after
again
against
age
agency
agent
ago
agree
agreement
ahead
air
all
allow
almost
alone
along
already
also
although
always
American
among
amount
analysis
and
animal
another
answer
any
anyone

U:--- 1000eng0rd.txt

Beginning of buffer

```
nyEngelskListe = []  
innfil = open("1000eng0rd.txt")  
for linje in innfil:  
    nyEngelskListe.append(linje.strip())  
  
print(nyEngelskListe)  
print("Antall ord i lista var", len(nyEngelskListe))  
print("Takk for nå")
```


a
ability
able
about
above
accept
according
account
across
act
action
activity
actually
add
address
administration
admit
adult
affect
after
again
against
age
agency
agent
ago
agree
agreement
ahead
air
all
allow
almost
alone
along
already
also
although
always
American
among
amount
analysis
and
animal
another
answer
any
anyone

U:--- 1000eng0rd.txt

Beginning of buffer

```
nyEngelskListe = []
innfil = open("1000eng0rd.txt")
for linje in innfil:
    nyEngelskListe.append(linje.strip())

print(nyEngelskListe)
print("Antall ord i lista var", len(nyEngelskListe))
print("Takk for nå")
```

- `s.strip()`
- `s.strip(chars)`

This method returns a new version of string *s* in which white space (blanks, tabs, and newlines) is removed from both ends (front and back) of *s*. If an argument is provided, characters in the string *chars* are removed instead of white space.

Parameter: *chars* A string specifying the characters to remove

Returns: A new version of the string

y', 'onto', 'open', 'operation', 'opportunity', 'option', 'or', 'order', 'organization', 'other', 'others', 'our', 'out', 'outside', 'over', 'own', 'owner', 'page', 'pain', 'painting', 'paper', 'parent', 'part', 'participa
nt', 'particular', 'particularly', 'partner', 'party', 'pass', 'past', 'patient', 'pattern', 'pay', 'peace', 'p
eople', 'per', 'perform', 'performance', 'perhaps', 'period', 'person', 'personal', 'phone', 'physical', 'pick'
, 'picture', 'piece', 'place', 'plan', 'plant', 'play', 'player', 'PM', 'point', 'police', 'policy', 'political'
, 'politics', 'poor', 'popular', 'population', 'position', 'positive', 'possible', 'power', 'practice', 'prepa
re', 'present', 'president', 'pressure', 'pretty', 'prevent', 'price', 'private', 'probably', 'problem', 'proce
ss', 'produce', 'product', 'production', 'professional', 'professor', 'program', 'project', 'property', 'protec
t', 'prove', 'provide', 'public', 'pull', 'purpose', 'push', 'put', 'quality', 'question', 'quickly', 'quite',
'race', 'radio', 'raise', 'range', 'rate', 'rather', 'reach', 'read', 'ready', 'real', 'reality', 'realize', 'r
eally', 'reason', 'receive', 'recent', 'recently', 'recognize', 'record', 'red', 'reduce', 'reflect', 'region',
'relate', 'relationship', 'religious', 'remain', 'remember', 'remove', 'report', 'represent', 'Republican', 'r
equire', 'research', 'resource', 'respond', 'response', 'responsibility', 'rest', 'result', 'return', 'reveal',
'rich', 'right', 'rise', 'risk', 'road', 'rock', 'role', 'room', 'rule', 'run', 'safe', 'same', 'save', 'say',
'scene', 'school', 'science', 'scientist', 'score', 'sea', 'season', 'seat', 'second', 'section', 'security',
'see', 'seek', 'seem', 'sell', 'send', 'senior', 'sense', 'series', 'serious', 'serve', 'service', 'set', 'seve
n', 'several', 'sex', 'sexual', 'shake', 'share', 'she', 'shoot', 'short', 'shot', 'should', 'shoulder', 'show'
, 'side', 'sign', 'significant', 'similar', 'simple', 'simply', 'since', 'sing', 'single', 'sister', 'sit', 'si
te', 'situation', 'six', 'size', 'skill', 'skin', 'small', 'smile', 'so', 'social', 'society', 'soldier', 'some
, 'somebody', 'someone', 'something', 'sometimes', 'son', 'song', 'soon', 'sort', 'sound', 'source', 'south',
'southern', 'space', 'speak', 'special', 'specific', 'speech', 'spend', 'sport', 'spring', 'staff', 'stage', 's
tand', 'standard', 'star', 'start', 'state', 'statement', 'station', 'stay', 'step', 'still', 'stock', 'stop',
'store', 'story', 'strategy', 'street', 'strong', 'structure', 'student', 'study', 'stuff', 'style', 'subject',
'success', 'successful', 'such', 'suddenly', 'suffer', 'suggest', 'summer', 'support', 'sure', 'surface', 'sys
tem', 'table', 'take', 'talk', 'task', 'tax', 'teach', 'teacher', 'team', 'technology', 'television', 'tell', 't
en', 'tend', 'term', 'test', 'than', 'thank', 'that', 'the', 'their', 'them', 'themselves', 'then', 'theory',
'there', 'these', 'they', 'thing', 'think', 'third', 'this', 'those', 'though', 'thought', 'thousand', 'threat'
, 'three', 'through', 'throughout', 'throw', 'thus', 'time', 'to', 'today', 'together', 'tonight', 'too', 'top'
, 'total', 'tough', 'toward', 'town', 'trade', 'traditional', 'training', 'travel', 'treat', 'treatment', 'tree'
, 'trial', 'trip', 'trouble', 'true', 'truth', 'try', 'turn', 'TV', 'two', 'type', 'under', 'understand', 'uni
t', 'until', 'up', 'upon', 'us', 'use', 'usually', 'value', 'various', 'very', 'victim', 'view', 'violence', 'v
isit', 'voice', 'vote', 'wait', 'walk', 'wall', 'want', 'war', 'watch', 'water', 'way', 'we', 'weapon', 'wear',
'week', 'weight', 'well', 'west', 'western', 'what', 'whatever', 'when', 'where', 'whether', 'which', 'while',
'white', 'who', 'whole', 'whom', 'whose', 'why', 'wide', 'wife', 'will', 'win', 'wind', 'window', 'wish', 'wit
h', 'within', 'without', 'woman', 'wonder', 'word', 'work', 'worker', 'world', 'worry', 'would', 'write', 'writ
er', 'wrong', 'yard', 'yeah', 'year', 'yes', 'yet', 'you', 'young', 'your', 'yourself']

Antall ord i lista var 1000

Takk for nå

>

```
1 # Funksjon som legger linjer (ord) fra en fil inn i ei liste
2 # Parameter: innfilnavn
3 # Returnerer lista med linjer fra innfilen, uten '\n' på slutten
4 def lagOrdlisteFraFil(filnavn):
5     liste = []
6     innfil = open(filnavn)
7     for linje in innfil:
8         liste.append(linje.strip())
9     return liste
10
```

```
# Del 2 Hovedprogram:
```

```
ordliste = lagOrdlisterFraFil("1001FraEngOrd.txt")
```

```
# Ordene i 1001FraEngOrd.txt er adskilt med strengen ' = '
```

```
# Bruker tekstfunksjonen split for å splitte opp linjene i 2 lister:
```

```
franskeOrd = []
```

```
engelskeOrd = []
```

```
for linje in ordliste:
```

```
    toOrd = linje.split(" = ")
```

```
    # toOrd[0] inneholder alle tegn før " = "
```

```
    # toOrd[1] inneholder alle tegn etter " = "
```

```
    franskeOrd.append(toOrd[0])
```

```
    engelskeOrd.append(toOrd[1])
```

```
# Filen 1000engOrd.txt inneholder 1000 engelske ord, ett ord pr. linje
```

```
nyEngelskliste = lagOrdlisterFraFil("1000engOrd.txt")
```

```
# Lage en liste med oversettelse til fransk for de ordene som finnes
```

```
# i begge de engelske listene
```

```
engelskTilFransk = []
```

```
for ord in engelskeOrd:
```

```
    if ord in nyEngelskliste:
```

```
        engelskTilFransk.append(ord + " = " + franskeOrd[engelskeOrd.index(ord)])
```

```
for linje in engelskTilFransk:
```

```
    print(linje)
```

```
print("Lengden til den nye lista engelskTilFransk er:" , len(engelskTilFransk))
```

```
français = French  
comme = as  
je = I  
son = his  
que = that  
il = he  
était = was  
pour = for  
sur = on  
sont = are  
avec = with  
ils = they  
être = be  
à = at  
un = one  
avoir = have  
ce = this  
à partir de = from  
par = by  
chaud = hot  
mot = word  
mais = but  
que = what  
certains = some  
est = is  
il = it  
vous = you  
ou = or  
eu = had  
la = the  
de = of  
à = to  
et = and  
un = a  
dans = in  
nous = we
```

```
# Del 2 Hovedprogram:
```

```
ordliste = lagOrdlisteFraFil("1001FraEngOrd.txt")
```

```
# Ordene i 1001FraEngOrd.txt er adskilt med strengen ' = '  
# Bruker tekstfunksjonen split for å splitte opp linjene i 2 lister:
```

```
franskeOrd = []  
engelskeOrd = []
```

```
for linje in ordliste:  
    toOrd = linje.split(" = ")  
    # toOrd[0] inneholder alle tegn før " = "  
    # toOrd[1] inneholder alle tegn etter " = "  
    franskeOrd.append(toOrd[0])  
    engelskeOrd.append(toOrd[1])
```

```
# Filen 1000engOrd.txt inneholder 1000 engelske ord  
nyEngelskliste = lagOrdlisteFraFil("1000engOrd.txt")
```

```
# Lage en liste med oversettelse til fransk for do  
# i begge de engelske listene
```

```
engelskTilFransk = []  
for ord in engelskeOrd:  
    if ord in nyEngelskliste:  
        engelskTilFransk.append(ord + " = " + franskeOrd[engelskeOrd.index(ord)])
```

```
for linje in engelskTilFransk:  
    print(linje)
```

```
print("Lengden til den nye lista engelskTilFransk er:" , len(engelskTilFransk))
```

- `s.split()`
- `s.split(sep)`
- `s.split(sep, maxsplit)`

This method returns a list of words from string *s*. If the substring *sep* is provided, it is used as the delimiter; otherwise, any white space character is used. If *maxsplit* is provided, then only that number of splits will be made, resulting in at most *maxsplit* + 1 words.

Parameters: *sep* A substring specifying the separator used for the split
maxsplit The maximum number of splits

Returns: A list of words that results from splitting the string

```
français = French  
comme = as  
je = I  
son = his  
que = that  
il = he  
était = was  
pour = for  
sur = on  
sont = are  
avec = with  
ils = they  
être = be  
à = at  
un = one  
avoir = have  
ce = this  
à partir de = from  
par = by  
chaud = hot  
mot = word  
mais = but  
que = what  
certains = some  
est = is  
il = it  
vous = you  
ou = or  
eu = had  
la = the  
de = of  
à = to  
et = and  
un = a  
dans = in  
nous = we
```

- `s.split()`
- `s.split(sep)`
- `s.split(sep, maxsplit)`

This method returns a list of words from string *s*. If the substring *sep* is provided, it is used as the delimiter; otherwise, any white space character is used. If *maxsplit* is provided, then only that number of splits will be made, resulting in at most *maxsplit* + 1 words.

Parameters: *sep* A substring specifying the separator used for the split
maxsplit The maximum number of splits

Returns: A list of words that results from splitting the string

```
# Del 2 Hovedprogram:
```

```
ordliste = lagOrdlisterFraFil("1001FraEngOrd.txt")
```

```
# Ordene i 1001FraEngOrd.txt er adskilt med strengen ' = '
```

```
# Bruker tekstfunksjonen split for å splitte opp linjene i 2 lister:
```

```
franskeOrd = []
```

```
engelskeOrd = []
```

```
for linje in ordliste:
```

```
    toOrd = linje.split(" = ")
```

```
    # toOrd[0] inneholder alle tegn før " = "
```

```
    # toOrd[1] inneholder alle tegn etter " = "
```

```
    franskeOrd.append(toOrd[0])
```

```
    engelskeOrd.append(toOrd[1])
```

```
# Filen 1000engOrd.txt inneholder 1000 engelske ord, ett ord pr. linje
```

```
nyEngelskliste = lagOrdlisterFraFil("1000engOrd.txt")
```

```
# Lage en liste med oversettelse til fransk for de ordene som finnes
```

```
# i begge de engelske listene
```

```
engelskTilFransk = []
```

```
for ord in engelskeOrd:
```

```
    if ord in nyEngelskliste:
```

```
        engelskTilFransk.append(ord + " = " + franskeOrd[engelskeOrd.index(ord)])
```

```
for linje in engelskTilFransk:
```

```
    print(linje)
```

```
print("Lengden til den nye lista engelskTilFransk er:" , len(engelskTilFransk))
```

- `s.split()`
- `s.split(sep)`
- `s.split(sep, maxsplit)`

This method returns a list of words from string `s`. If the substring `sep` is provided, it is used as the delimiter; otherwise, any white space character is used. If `maxsplit` is provided, then only that number of splits will be made, resulting in at most `maxsplit + 1` words.

Parameters: `sep` A substring specifying the separator used for the split
`maxsplit` The maximum number of splits

Returns: A list of words that results from splitting the string

```
français = French  
comme = as  
je = I  
son = his  
que = that  
il = he  
était = was  
pour = for  
sur = on  
sont = are  
avec = with  
ils = they  
être = be  
à = at  
un = one  
avoir = have  
ce = this  
à partir de = from  
par = by  
chaud = hot  
mot = word  
mais = but  
que = what  
certains = some  
est = is  
il = it  
vous = you  
ou = or  
eu = had  
la = the  
de = of  
à = to  
et = and  
un = a  
dans = in  
nous = we
```

Skrive til en tekstfil

- Først åpne en fil for skriving:

```
min_fil = open("mittUtFilNavn.txt", "w")
```

- Deretter skrive tekst (en streng-verdi) til slutten av filen:

```
min_fil = open("mittUtFilNavn.txt", "w")
```

```
min_fil.write("Dette havner i filen")
```

- Man vil typisk ha linjeskift i det man skriver ut:

```
min_fil.write("Dette havner i filen\n")
```

```
min_fil.write("Dette havner på\n ulike\n linjer\n")
```

- Når man er ferdig bør man lukke filen:

```
min_fil.close()
```



```
# Fra før har vi de to listene franskeOrd og engelskeOrd
# Filen 1000engOrd.txt inneholder 1000 engelske ord, ett ord pr. linje
nyEngelskliste = lagOrdlsteFraFil("1000engOrd.txt")

# Lage en liste med oversettelse til fransk for de ordene som finnes
# i begge de engelske listene

engelskTilFransk = []
for ord in engelskeOrd:
    if ord in nyEngelskliste:
        engelskTilFransk.append(ord + " = " + franskeOrd[engelskeOrd.index(ord)])

# Skrive den nye ordboka til fil:

utfil = open ("630EngTilFra.txt", "w") # legg merke til 2. parameter 'w' for write
for linje in engelskTilFransk:
    utfil.write(linje)

utfil.close() # Lukker filen. Dette forhindrer ytterligere skriving til den
print("Filen 630engtilfra.txt skrevet. Avslutter...")
```

```
# Fra før har vi de to listene franskeOrd og engelskeOrd
# Filen 1000engOrd.txt inneholder 1000 engelske ord, ett ord pr. linje
nyEngelskliste = lagOrdlisterFraFil("1000engOrd.txt")

# Lage en liste med oversettelse til fransk for de ordene som finnes
# i begge de engelske listene

engelskTilFransk = []
for ord in engelskeOrd:
    if ord in nyEngelskliste:
        engelskTilFransk.append(ord + " = " + franskeOrd[engelskeOrd.index(ord)])

# Skrive den nye ordboka til fil:

utfil = open ("630EngTilFra.txt", "w") # legg merke til 2. parameter 'w' for write
for linje in engelskTilFransk:
    linjeTilFil = linje + '\n' # legger til tegn for linjeskift
    utfil.write(linjeTilFil)

utfil.close() # Lukker filen. Dette forhindrer ytterligere skriving til den
print("Filen 630engtilfra.txt skrevet. Avslutter...")
```